Inversio-ongelmien laskennallinen peruskurssi 2012

MATLAB Exercise 8

Statistical Linear Inverse Problem

Consider equation

$$\begin{aligned} &Y = AX + E, \\ &E \sim N(0, \Gamma_n), \quad \Gamma_n = C_n C_n^T, \\ &X \sim N(x_0, \Gamma_{pr}), \quad \Gamma_{pr} = C_{pr} C_{pr}^T. \end{aligned}$$

The *a posteriori* density of X:

$$X \sim N(\overline{x}, \Gamma_{post}),$$

where

$$\overline{x} = x_0 + \Gamma_{pr} A^T (A \Gamma_{pr} A^T + \Gamma_n)^{-1} (y - A x_0 - e_0), \tag{1}$$

$$\Gamma_{post} = \Gamma_{pr} - \Gamma_{pr} A^T (A \Gamma_{pr} A^T + \Gamma_n)^{-1} A \Gamma_{pr}; \tag{2}$$

or

$$\Gamma_{post} = (\Gamma_{pr}^{-1} + A^T \Gamma_n A)^{-1} \tag{3}$$

$$\overline{x} = \Gamma_{post} (A^T \Gamma_n^{-1} (y - e_0) + \Gamma_{pr} x_0). \tag{4}$$

Note that the expectation value $\overline{x}$ in equation (4) can also be calculated by solving the overdetermined problem

$$\underbrace{\begin{pmatrix} C_n^{-1} y \\ C_{pr}^{-1} x_0 \end{pmatrix}}_{b} = \underbrace{\begin{pmatrix} C_n^{-1} A \\ C_{pr}^{-1} \end{pmatrix}}_{B} \overline{x}, \tag{5}$$

and

$$\Gamma_{post} = (B^T B)^{-1}.$$

If instead of *a priori* density we have an *a priori* model

$$x_0 = LX + E_{pr}, \quad E_{pr} \sim N(0, \Gamma_{pr}), \quad \Gamma_{pr} = C_{pr} C_{pr}^T,$$

the equation (5) has the form

$$\underbrace{\begin{pmatrix} C_n^{-1} y \\ C_{pr}^{-1} x_0 \end{pmatrix}}_{b} = \underbrace{\begin{pmatrix} C_n^{-1} A \\ C_{pr}^{-1} L \end{pmatrix}}_{B} \overline{x}.$$

Note also, that the *a priori* model

$$x_0 = LX + E_{pr}, \quad E_{pr} \sim N(0, \Gamma_{pr}), \quad \Gamma_{pr} = C_{pr} C_{pr}^T,$$

can be presented as a Gaussian prior as

$$X \sim N(x_0', \Gamma_{pr}'),$$

where

$$\begin{aligned} x_0' &= L^{-1} x_0, \\ \Gamma_{pr}' &= L^{-1} \Gamma_{pr} (L^{-1})^T. \end{aligned}$$

If the prior is given as

$$X \sim N(x_0, \Gamma_{pr}), \quad \Gamma_{pr} = CC^T$$

it can be sampled (i.e. calculate realizations of it) by calculating

$$x = Cw + x_0, \quad w \sim N(0, I).$$

A priori model

$$x_0 = LX + E_{pr}, \quad E_{pr} \sim N(0, \Gamma_{pr}), \quad \Gamma_{pr} = C_{pr} C_{pr}^T,$$

can be sampled by calculating

$$x = L^{-1}(x_0 - C_{pr} w), \quad w \sim N(0, I).$$

Exercises

1. Construct signal s and convolution matrix A with convolution kernel a as follows:

```
x = linspace(0,1,500);
s = (x+5).*(x-2).*(x-0.3).*(x-0.6).*(x-0.9);
a = 0:99;
a = [a a(100:-1:1)];
a = a/sum(a);
A = conmatrix(a,500);
```

   Plot signal s and the kernel a.

2. Construct noisy data by calculating the convolution of signal s and the kernel a and adding some random Gaussian noise n_g, where

```
n_g = c * randn(size(data)),
```

   and $c \in (0.01, 0.1)$ (choose yourself!). Calculate the variance of error by

```
evar = var(data - noisy_data)
```

   Note that in practice we do not know the noiseless measurement and the variance must be estimated.

3. Write MATLAB functions `L = L1(n)` and `L = L2(n)` that construct first and second order difference matrices of size $n \times n$, respectively. Here

$$L1 = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{pmatrix} \quad L2 = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}$$

   Sample prior model (using different variances δ)

$$0 = LX + E, \quad E \sim N(0, \delta I),$$

   where L is

   a) white noise (identity matrix),
   b) first order difference matrix,
   c) second order difference matrix.

4. Solve deconvolution problem of Problem 1 as a statistical inverse problem

$$y = AX + E_n, \quad E_n \sim N(0, \delta_n I), \tag{6}$$

$$0 = LX + E_{pr}, \quad E_{pr} \sim N(0, \delta_{pr} I), \tag{7}$$

   where δ_n is the error variance of the noisy measurement, and L is

   a) white noise (identity matrix),

   b) first order difference matrix,

   c) second order difference matrix.

   Try different *a priori* variances δ_{pr}.

   Plot the calculated solution, and plot also 95% Bayesian credibility set, i.e. in this Gaussian case, $\overline{x} \pm 2\delta_{post}$, where δ_{post} is the *a posteriori* standard deviation which is the square root of the diagonal of the posterior covariance matrix.

5. Same as Problem 4, but this time set the *a priori* variances of the end-points *much* larger, for example

```
delta(1) = 100000 * delta(1);
delta(end) = 100000 * delta(end);
```